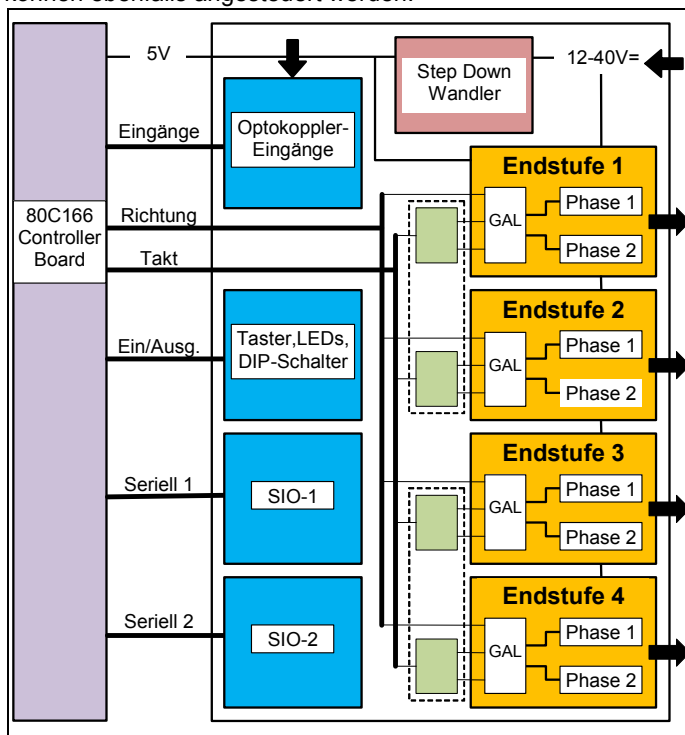


Schritt-MOTOR-Steuerung SMOST

Wer hat nicht dem ein oder anderen Schrittmotor mit mehr oder weniger Erfolg Drehbewegungen zu entlocken versucht. Dabei ist die Schrittmotor-Endstufe das kleinere Problem, es scheitert oft an einer geeigneten Ansteuerung. Die hier vorgestellte Schrittmotor-Karte „SMOST“ als Erweiterung zum 80C166-Controller-Board kann bis zu vier 2-Phasen-Schrittmotoren gleichzeitig steuern. Mit der entsprechenden Software liefert das Controller-Board alle erforderlichen Impulse und reagiert auf Informationen von Tastern, Sensoren, Schnittstellen, usw. Andere Schrittmotor-Karten mit Takt- und Richtungs-Eingang können ebenfalls angesteuert werden.



Blockschaltbild SMOST-Board

Technische Daten SMOST-Board

- 4 Schrittmotor-Endstufen mit jeweils
 - 2 bipolare Ausgangspaare, max. 0,5 A Phasenstrom
 - Strombegrenzung in 3 Stufen oder über Potentiometer
 - Cut-Off-Time 35 μ sek, Stromabsenkung bei Stillstand
 - Voll- und Halbschritt, Schritte über GAL veränderbar
 - LED-Anzeigen für Takt und Richtung
- 12 Optokoppler-Eingänge (26-pol. Pfosten)
- 4 x Nullpunktsensor, 8 x Positionssensor
- 2 Serielle Schnittstellen (14-pol. Pfosten)
 - 1 x PC-Anschluß (9600 Baud, 8 Bit, 1 Stop)
 - 1 x Reserve (Monitor-Programm)
- 4 Ein-/Ausgänge, +5V, GND (14-pol. Pfosten) Reserve
- 4-poliger Dip-Schalter für Programm-Einstellungen
- 10 Taster für manuelle Steuerung, 1 Reset-Taster
- 4 LED-Anzeigen für Betriebszustand

Versorgung 12 - 40 V Gleichspannung, ca. 2 A
Step-Down Wandler für 5 V Versorgung, max. 1 A

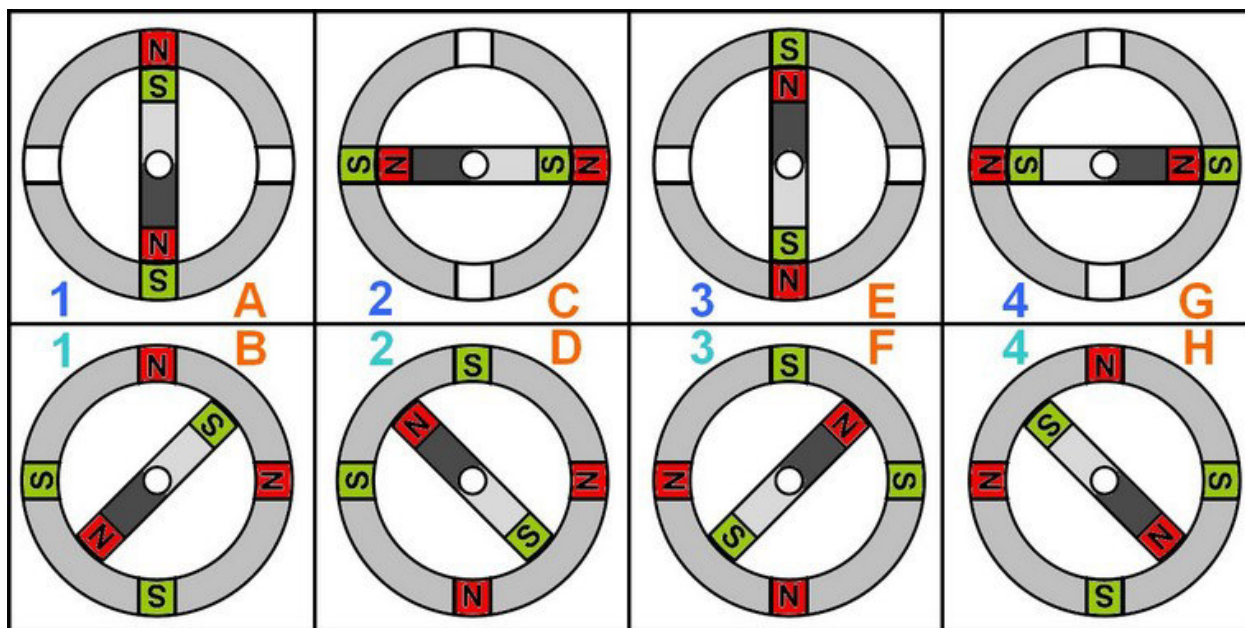
Erforderliche Bestückung auf dem 80C166-Board:
64 kB RAM, EPROM mit SMOST-Software, EEPROM.
Die 80C166-Pinbelegung für SMOST ist in Tabelle 6.

Zum besseren Verständnis von SMOST sollten zuerst einige prinzipielle Begriffe und Funktionen des Schrittmotors und seiner Ansteuerung geklärt werden.

Schrittmotor-Begriffe und -Funktionen

Ein Takt alleine bewegt noch keinen Schrittmotor. Jeder Schrittmotor benötigt eine Elektronik, die aus dem Takt einen Ablauf mit entsprechenden Strömen und Stromrichtungen (Polaritäten) generiert. Betrachten wir als Beispiel einen Schrittmotor mit 2 um 90 Grad versetzte Polwicklungen (Phasen). Werden die Phasen bestromt, entsteht ein Magnetfeld. Der Rotor bewegt sich mit seinem Magnetpolpaar in eine Position, wo zwischen Rotor und Phase der Luftspalt am kleinsten und der Magnetfluß am größten ist (zur Erinnerung: ungleiche Pole ziehen sich an, gleiche stoßen sich ab).

Durch Ändern der Bestromungsrichtung kann der Rotor vier verschiedene Positionen einnehmen. Polaritätsänderungen führen in einer bestimmten Reihenfolge zu einer Drehbewegung. Nach jedem Takt müssen sich die Polaritäten ändern. Die Rotorpositionen sind in den folgenden Abbildungen dargestellt:



Schritt-MOTOR-Steuerung SMOST

Bei der ersten Methode wird nur 1 Phase bestromt (1 - 4):

Tabelle 1

Rechts-Drehung					Links-Drehung				
Takt	Phase 1 Pol./Strom		Phase 2 Pol./Strom		Takt	Phase 1 Pol./Strom		Phase 2 Pol./Strom	
1	0	1	X	0	1	0	1	X	0
2	X	0	0	1	2	X	0	1	1
3	1	1	X	0	3	1	1	X	0
4	X	0	1	1	4	X	0	0	1

Eine **Sequenz** ist eine „elektrische“ Umdrehung von 360 Grad (elektrischer Schrittwinkel), die notwendig ist, um einen kompletten mechanischen Schrittwinkel durchzuführen.

Bei **Vollschritt** besteht jede Sequenz aus 4 Takten und im Beispiel-Motor einem daraus resultierenden mechanischen Schrittwinkel von 360 Grad, was hier einer Umdrehung entspricht.

Aus den beiden Tabellen ist ersichtlich, bei welcher Reihenfolge sich die Drehrichtung ändert. Ein Richtungswechsel ergibt sich auch bei umgekehrtem Anschluß von einer oder beiden Phasen.

Ein Schrittmotor mit nur 4 Positionen liefert keine saubere Drehung; es sind mehr Rotorpositionen notwendig.

Eine zweite Methode ist das Bestromen von 2 Phasen (1 - 4):

Tabelle 2

Rechts-Drehung					Links-Drehung				
Takt	Phase 1 Pol./Strom		Phase 2 Pol./Strom		Takt	Phase 1 Pol./Strom		Phase 2 Pol./Strom	
1	0	1	0	1	1	0	1	0	1
2	1	1	0	1	2	0	1	1	1
3	1	1	1	1	3	1	1	1	1
4	0	1	1	1	4	1	1	0	1

Kombiniert man die beiden Vollschritt-Methoden, erhält man die **Halbschritt**-Steuerung (A - H).

Tabelle 3

Rechts-Drehung					Links-Drehung				
Takt	Phase 1 Pol./Strom		Phase 2 Pol./Strom		Takt	Phase 1 Pol./Strom		Phase 2 Pol./Strom	
A	0	1	X	0	A	0	1	X	0
B	0	1	0	1	B	0	1	0	1
C	X	0	0	1	C	X	0	1	1
D	1	1	0	1	D	0	1	1	1
E	1	1	X	0	E	1	1	X	0
F	1	1	1	1	F	1	1	1	1
G	X	0	1	1	G	X	0	0	1
H	0	1	1	1	H	1	1	0	1

Im **Halbschritt**-Betrieb besteht eine Sequenz aus 8 Takten.

Auch hiermit wird der Beispiel-Motor noch „ruckeln“, besonders bei kleinen Drehzahlen.

Eine weitere Verbesserung ist möglich, indem die Bestromung nicht ein- und ausgeschaltet, sondern stufenweise gesenkt bzw. erhöht wird. Durch diesen **Mikroschritt**-Betrieb entstehen ruhige Drehbewegungen. Für einen Schrittmotor mit nur 4 Rotorpositionen ist diese Lösung allerdings nicht praktikabel; die Leistung, also das **Drehmoment**, nimmt stark ab und die Positionier-Genauigkeit geht verloren.

Durch mehr Polwicklungen und Rotor-Magnetpolpaare erhält man mehr Rotorpositionen.

Im Allgemeinen besitzen Schrittmotoren viele Polwicklungen, die als 2 und mehr Phasen ausgeführt sind, teilweise mit Mittelanzapfung. Auf den Rotoren befinden sich mehrere Magnetpolpaare entweder auf einem Anker, der wie ein sehr breites Zahnrad ausgebildet ist, oder auf einer Scheibe.

Die Anzahl der Motor-Winkelschritte kann durch langsames Drehen von Hand ermittelt werden. Es ist ein mehr oder weniger starkes „Einrasten“ zu spüren.

In alten Festplattenlaufwerken befinden sich für die präzise Kopfpositionierung Schrittmotoren mit 200 und mehr (mechanischen) Winkelschritten. Dazu 3 Berechnungen:

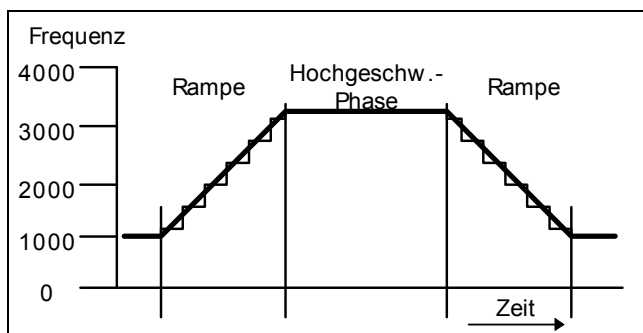
Schrittwinkel $360 \text{ Grad} / 200 \text{ Winkelschritte} = 1,8 \text{ Grad}$
 Takte pro Umdrehung bei Vollschritt $200 \text{ Winkelschritte} \times 4 \text{ Takte} = 800 \text{ Takte}$
 Takte pro Umdrehung bei Halbschritt $200 \text{ Winkelschritte} \times 8 \text{ Takte} = 1600 \text{ Takte}$

Da ein Schrittmotor mit 2 Phasen und 1 Magnetpolpaar im Halbschritt 8 Takte pro Umdrehung benötigt und der Festplatten-Schrittmotor 1.600 Takte, erhalten wir über die Taktfrequenz keine Aussage zur Drehzahl, jedoch einen Anhaltspunkt für die Positioniergenauigkeit. Ist eine hohe Drehzahl gefordert, sollte ein Schrittmotor mit größeren Winkelschritten eingesetzt werden.

Bei Einschalten einer Taktfrequenz von 10.000 Hz dreht sich ein Schrittmotor nicht. Der Rotor ist zu träge, um dem schnell rotierenden Stator-Magnetfeld zu folgen.

Mit einer sogenannten **Start-/Stopfrequenz**, die je nach Schrittmotor und Belastung zwischen 50 und 2.000 Hz liegt, ist ein sicherer Anlauf gewährleistet. Nach dem Start kann die Taktfrequenz erhöht werden. Diese Erhöhung bestimmt die **Beschleunigung** des Motors und darf beliebig langsam, jedoch nicht beliebig schnell erfolgen; der Schrittmotor steigt sonst aus.

Ein Überschreiten der maximalen Taktfrequenz bzw. Drehzahl führt ebenfalls zum sofortigen Stillstand.



Bei „rechner-erzeugten“ Rampen wird pro Zeiteinheit die Frequenz um den Beschleunigungs-Wert verändert. Dadurch entsteht eine leichte Stufenform, die im Mittel jedoch annähernd linear ist.

Zur Nullpunktsuche oder für langsame Fahrten wird eine Taktfrequenz gewählt, bei der sich der Schrittmotor ohne Rampen sicher in beide Richtungen dreht.

Nach dieser kleinen Einführung zurück zu SMOST.

Schritt-MOTOR-Steuerung SMOST

SMOST Hardware

Jede der 4 Endstufen besteht aus 2 Schrittmotor-IC's PBL 3717, einem GAL 16V8 und einem retriggerbaren monostabilen Multivibrator 74HC123.

Der Ausgang des 74HC123 geht in seinen Ruhezustand, wenn ca. 35 bis 50 mSek kein Taktimpuls anliegt. Mit diesem Signal kann der Motorstrom im Stillstand abgesenkt werden.

Das GAL bestimmt Schrittfolge und Bestromung; die Ausgänge führen zu den beiden Schrittmotor-IC's.

Diese erzeugen mit nur wenigen externen Bauteilen die Phasenströme für die Schrittmotoren. Die Ausgänge ändern ihre Polarität entsprechend dem TTL-Eingangsspiegel an Pin 8 (PH). Zwei weitere TTL-Eingänge, Pin 9 und 7 (I0 und I1), stellen eine interne Komparator-Schwelle auf ca. 420, 250, 80 oder 0 mV ein. Über den 1 Ohm-Widerstand Rx1 (Rx6) am Pin 16 des PBL 3717 entsteht bei Stromfluß ein Spannungsabfall, der über Rx2 (Rx7) an den Komparator-Eingang Pin 10 gelangt und bei Überschreitung der eingestellten Komparator-Schwelle den Strom abschaltet. Nach ca. 35 µSek „Cut-Off-Time“, bestimmt durch Rx3 (Rx8) und Cx3 (Cx8), wird der Strom wieder eingeschaltet und der Vorgang beginnt von vorne. Die Endstufe liefert nur so lang (oder kurz) den eingestellten Strom, wie zum Aufbau bzw. Erhalt des Magnetfeldes in den Phasen erforderlich ist (Zur Erinnerung: ein hoher induktiver Spulen-Widerstand sinkt nach dem Aufbau des Magnetfeldes auf den geringen ohmschen Spulen-Widerstand). Alternativ kann der Strom auch über ein Potentiometer geregelt werden. Über den Spannungsteiler Rx4 (Rx5) / Rx2 (Rx7) liegt am Komparator-Eingang eine „Vorspannung“, die bei Erhöhung durch den Spannungsabfall am 1 Ohm-Widerstand ebenfalls zur Überschreitung der Komparator-Spannung und somit zum Abschalten des Stroms führt. Nachteil: eine Steuerung der Bestromung ist nicht mehr möglich, da die beiden „Strom“-Eingänge des PBL 3717 bei dieser Alternative auf Low-Pegel liegen müssen.

Über 3 Jumper an den Eingängen der GALs kann ein Ablauf beeinflusst werden:

1	Schrittwahl	1=Voll	0=Halb
2	Phasen-Strom	1=60%	0=100%
3	Stillstand-Strom	1=60%	0=100%

Andere Varianten sind durch Änderung der GALs möglich.

4 Optokoppler-Eingänge stehen für Sensoren oder Kontakte mit Schließer-Funktion für eine Nullpunktsuche zur Verfügung.

Bei der Nullpunktsuche fährt der Schrittmotor erst in Richtung des Sensors bis dieser einschaltet, wartet 0,2 Sekunden und fährt dann noch in die entgegengesetzte Richtung, bis der Sensor wieder ausschaltet. Das Auffinden des Nullpunktes wird quittiert. Beim Anschluß eines Schrittmotors ist darauf zu achten, daß er sich bei der Nullpunktsuche in Richtung Sensor dreht.

Das Aktivieren einer der übrigen 8 Optokoppler-Eingänge führt jeweils zu einer Meldung. Damit kann das Erreichen bestimmter Positionen oder Zustände überwacht werden.

SMOST Software

Wie bereits angedeutet, werden alle Takte und Richtungs-Signale auf dem 80C166-Board mit der Software erzeugt. Jeder Takt wird in einem 32-Bit-Zähler registriert und kann als „Aktuelle Position“ abgefragt oder verändert werden. Nach einer Nullpunktsuche wird dieser Wert auf Null gesetzt. Bei dem Kommando „Go“ erstellt der Controller aus den Parametern „Untere Frequenz“, „Obere Frequenz“ und „Beschleunigung“ eine Rampentabelle.

Die Takte werden mit diesen Werten für die steigende Rampe bis zur Hochgeschwindigkeitsphase ausgegeben. Ist der Dauerlaufmodus eingeschaltet, erfolgt die weitere Taktausgabe mit dem Parameter „Obere Frequenz“. Bei einem „Halt“-Kommando wird die Rampentabelle in umgekehrter Reihenfolge für die fallende Flanke bis zum Stillstand abgearbeitet. Ist der Positionier-Modus eingeschaltet, wird in der Tabelle zusätzlich die Differenz zwischen Aktueller Position und End-Position berücksichtigt sowie die Schrittzahl für die Hochgeschwindigkeitsphase ermittelt. Werden für die beiden Rampen mehr Schritte benötigt, als für die Differenz von Aktueller Position und End-Position erforderlich wären, kann die Hochgeschwindigkeitsphase nicht erreicht werden. Da nach dem Start alle erforderlichen Taktfrequenzen und Schritte berechnet sind, dürfen alle Parameter (ausgenommen der Dauerlauf- oder Positionier-Modus) verändert werden. Ein laufender Schrittmotor kann nur mit dem Kommando „Halt“ vorzeitig angehalten werden. Nach einem „Go“ kann sogar unmittelbar das Kommando „Nullpunkt suchen“ gegeben werden; nach Erreichen der End-Position sucht der Schrittmotor dann sofort seinen Nullpunkt. Interessant ist noch der Parameter „Zwischen-Position“. Immer wenn die Aktuelle Position gleich diesem Wert ist, sendet das Programm eine Meldung (z1...z4) über die serielle Schnittstelle. Dieser Parameter kann zu jeder Zeit und beliebig oft geändert werden. Eine Meldung wird auch gesendet, wenn Endposition (e1...e4) oder Nullpunkt (n1...n4) erreicht sind, bei Verlassen des Nullpunkts (v1...v4) oder bei Aktivieren der 8 Sensor-Eingänge (s1...s8).

Das Kommando „Nullpunkt suchen“ führt nur dann zu einer Suche, wenn ein Nullpunkt-Sensor vorhanden ist und der Parameter „Ja, mit Nullpunktsensor“ gesetzt ist. Bei „Kein Nullpunktsensor“ wird nur die Aktuelle Position auf Null gesetzt und die Meldung (n1...n4) ausgegeben. Alle Kommandos und Meldungen sind in Tabelle 5 aufgelistet.

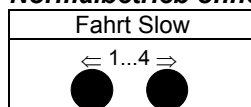
Schritt-MOTOR-Steuerung SMOST

SMOST Tasterfunktionen

Mit den Tastern können Parameter verändert und Schrittmotoren manuell gesteuert werden. Eine bessere Methode zur Parametereingabe und Steuerung ist der Anschluß eines PCs an die serielle Schnittstelle. Über die Schnittstelle werden Parameter und Kommandos empfangen sowie Informationen und Meldungen gesendet.

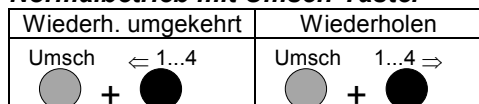
Bei SMOST wird zwischen Normalbetrieb und Programmiermodus unterschieden.

Normalbetrieb ohne Umsch-Taster



Bei Betätigen der Taster 1...4 links oder 1...4 rechts dreht sich der Schrittmotor mit der eingestellten Taktfrequenz in die angewählte Richtung, bis der Taster losgelassen wird.

Normalbetrieb mit Umsch-Taster

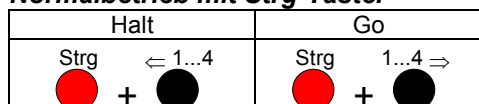


Bei Betätigen der Taster 1...4 links fährt der Schrittmotor die letzte Schrittzahl in umgekehrter Richtung (= Kommando „X Wiederholen umgekehrt“).

Bei Betätigen der Taster 1...4 rechts fährt der Schrittmotor die letzte Schrittzahl in der letzten Richtung (= Kommando „Wiederholen“).

Die Wirkungsweise des Umsch-Tasters kann über den Dipschalter 2 umgekehrt werden.

Normalbetrieb mit Strg-Taster

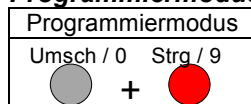


Mit Taster 1...4 links wird ein Schrittmotor angehalten (= Kommando „Halt“).

Mit Taster 1...4 rechts wird ein Schrittmotor gestartet (= Kommando „Go“).

Voraussetzung zum Starten ist, daß Aktuelle Position und End-Position ungleich sind.

Programmiermodus ein- / ausschalten



Um in den Programmiermodus zu wechseln den Umsch-Taster gedrückt halten und den Strg-Taster betätigen; die gelbe LED beginnt zu blinken.

Zum Ausschalten oder Abbrechen die Tasterkombination wiederholen; die gelbe LED geht aus.

Bei einer Zifferneingabe hat der Umsch-Taster die Ziffer 0 und der Strg-Taster die Ziffer 9.

Nach einer kompletten oder falschen Eingabe wird der Programmier-Modus automatisch beendet.

Programmiermodus

Im Programmier-Modus haben die Taster unterschiedliche Funktionen (siehe Tabelle 4).

Umsch / 0 Strg / 9 	0...9 	← 1...4 → 	0 ... 9 	ACHTUNG ! Punkte NICHT mit eingeben
Funktion	Taster	Motor	Wert = Tasterfolge	Seriell
Takt-Frequenz	1	M 1 - 4	00.050 - 05.000	T 50 - 5.000
Untere Frequenz	2	M 1 - 4	00.050 - 05.000	U 50 - 5.000
Obere Frequenz	3	M 1 - 4	00.050 - 20.000	O 50 - 20.000
Beschleunigung	4	M 1 - 4	00.005 - 01.000	B 5 - 1.000
Aktuelle Position	5	M 1 - 4	00.000 - 50.000	A -2.147.483.648 - +2.147.483.648
End-Position	6	M 1 - 4	00.000 - 50.000	E -2.147.483.648 - +2.147.483.648
Dauerlauf-Modus	7	M 1 - 4	-	D 1 - 4
Positionier-Modus	8	M 1 - 4	-	P 1 - 4
Halt (alle Motoren)	9	-	-	H 9
Nullpunkt suchen	0	M 1 - 4	-	N 1 - 4

Tabelle 4: Tasterfunktionen im Programmiermodus

Schritt-MOTOR-Steuerung SMOST

SMOST Kommunikation

Die Kommunikation mit SMOST erfolgt über die serielle Schnittstelle mit 9.600 Baud / 8 Bit / 1 Stopbit / No Parity. Mit einem PC-Terminal-Programm (ASCII-Übertragung) ist eine Steuerung und Abfrage von SMOST möglich.

Für komplexe Schrittmotor-Abläufe ist jedoch ein Programm erforderlich, das Kommandos erzeugt und Meldungen auswertet.

SMOST empfängt alle Kommandos als Buchstabe oder ganzes Wort ohne Ziffern und Sonderzeichen. Zwischen Groß- und Kleinschreibung wird nicht unterschieden:

M1 / m2 / M 3 / m 4 / Motor 2

Mehrere Kommandos in einer Zeile sind zulässig; sie müssen durch Komma oder Doppelpunkt getrennt sein:

m1, t500 / M1, T500 / M1:T500:M2:T800 /
Motor 1, Taktfrequenz 500

Nach einem Semikolon dürfen bis zum Zeilen-Ende Kommentare hinzugefügt werden:

M1,T500 ; Taktfrequenz 500 Hz für Motor 1 <ret>

Es gibt 3 Typen von Meldungen:

- 1 Großbuchstaben Status-Meldungen
- 2 Kleinbuchstaben Positions-Meldungen
- 3 Semikolon Infos und Kommentare

Alle Kommandos und Meldungen sind in Tabelle 5 zusammengefaßt.

Kommando	Parameter	Beschreibung
? Hilfe	-	Hilfe abfragen
A * Aktuelle Position	±2.147.483.648	Aktuelle Position setzen / abfragen
B * Beschleunigung	5 - 1.000	Motor-Beschleunigung setzen / abfragen (Takte pro ms)
C Copy Data	-	Daten von Speicher in EEPROM kopieren
D Dauerlauf-Modus	0 / 1 - 4 / 9	Dauerlauf-Modus ein (Positionier-Modus aus)
E * End-Position	±2.147.483.648	End-Position setzen / abfragen
e Meldung	1 - 4	Meldung „End-Position erreicht“
F Fahrt Slow	0 / 1 - 4 / 9	Fahrt mit Takt-Frequenz (Slow) in angewählter Richtung
G Go	0 / 1 - 4 / 9	Motor starten / End-Position anfahren
H Halt	0 / 1 - 4 / 9	Motor anhalten
I Info	-	Motor-Nummer ... 1 2 3 4 Takt-Frequenz = 500 500 200 1000 Untere Frequenz= 1000 500 100 100 Obere Frequenz= 10000 8000 5000 5000 Beschleunigung = 500 20 100 10 Akt Position = 0 -6540 0 0 End-Position = 3500 732478 0 0 Zwischen-Pos = 1750 0 0 0 Positionier-Mod= 1 1 1 0 Nullpunkt-Suche= 1 0 0 0 Akt. Motor = 3
J Ja, mit Nullpunkt	0 / 1 - 4 / 9	Nullpunkt-Sensor anfahren und akt. Position auf 0 setzen
K Kein Nullpunkt	0 / 1 - 4 / 9	Kein Nullpunkt-Sensor; aktuelle Position direkt auf 0 setzen
L Linkslauf	0 / 1 - 4 / 9	Linkslauf bei Fahrten mit Takt-Frequenz (Slow)
M Motor aktuell *	1 - 4	Motor-Nr. für nachfolgende Kommandos setzen / abfragen
N Nullpunkt suchen	0 / 1 - 4 / 9	Nullpunkt suchen mit Takt-Frequenz
n Meldung	1 - 4	Meldung „Nullpunkt erreicht“
O * Obere Frequenz	50 - 20.000 Hz	Obere Frequenz (Fast) setzen / abfragen
P Positionier-Modus	0 / 1 - 4 / 9	Positionier-Modus ein (Dauerlauf-Modus aus)
q Meldung	0 - 9	Quittierung (q0) und Fehlermeldungen (q1-q9)
R Rechtslauf	0 / 1 - 4 / 9	Rechtslauf bei Fahrten mit Takt-Frequenz (Slow)
S Status	- / 0 / 1 - 4 / 9	Status abfragen M1, T500, U1000, O10000, B500, A0, E3500, Z1750, P1, J1 M2, T500, U500, O8000, B20, A-6540, E732478, Z0, P1, J0 M3,
s Meldung	1 - 8	Meldung „Sensor erreicht“
T * Takt-Frequenz	50 - 5.000 Hz	Takt-Frequenz (Slow) setzen / abfragen
U * Untere Frequenz	50 - 5.000 Hz	Untere Frequenz (Start/Stop) setzen / abfragen
V Versions-Nr.	-	Versions-Nummer des Programms abfragen
v Meldung	1 - 4	Meldung „Verlassen des Nullpunkts“
W Wiederholen	0 / 1 - 4 / 9	Wiederholen letzte Schrittzahl in letzter Richtung
X Wiederh. umgekehrt	0 / 1 - 4 / 9	Wiederholen letzte Schrittzahl in umgekehrter Richtung
Z * Zwischen-Position	±2.147.483.648	Zwischen-Position setzen / abfragen
z Meldung	1 - 4	Meldung „Zwischen-Position erreicht“
	Zulässige Werte:	0 = aktueller Motor (mit M1 - M4 vorher gesetzt) 1 - 4 = angegebener Motor 9 = alle Motoren - = Zustand abfragen

Tabelle 5: Kommandos und Meldungen für Schritt-Motor-Steuerung SMOST